

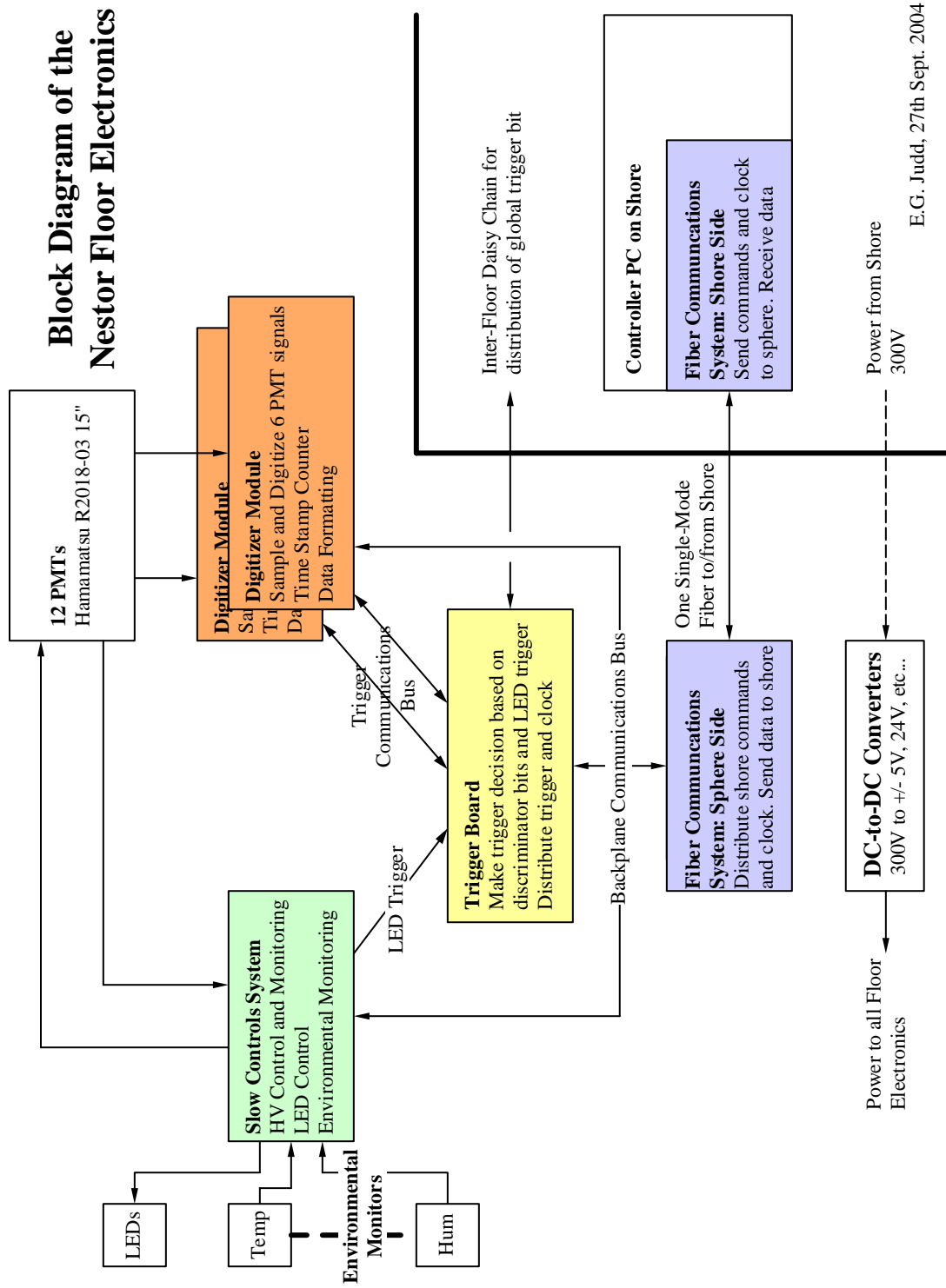
## **Appendix 1 – Nestor Electronics Details**

Nestor Sphere Block Diagram .....	4
Requirements and Justification for the Nestor Digitizer Module (NDM) .....	5
1. Physical Characteristics .....	5
1.1. Form Factor.....	5
1.2. Temperature .....	5
1.3. Atmosphere .....	5
1.4. Power .....	5
2. Connections.....	5
2.1. Backplane Communications Bus (BCB) .....	5
2.2. Trigger Board (TB).....	6
2.3. PMT signals .....	6
3. Control Functionality .....	6
3.1. Global Clock from TB .....	6
3.2. Local Oscillator.....	6
3.3. Clock Source Selection Jumper .....	6
3.4. Trigger Word from TB.....	6
3.5. Logic Control.....	6
3.6. Time Stamp Counter .....	7
3.7. Event Data Gathering and Formatting .....	7
3.8. Scaler Data Gathering and Formatting .....	7
3.9. Storage .....	7
3.10. “Busy” Signal to TB .....	7
3.11. Data Transfer to FCS .....	7
4. Digitization Circuitry .....	8
4.1. Ground and Power Separation .....	8
4.2. Termination of PMT signals .....	8
4.3. Noise .....	8
4.4. Comparison of PMT Signals to Threshold .....	8
4.5. Scalers for Comparator Output .....	8
4.6. Stretching of Comparator Outputs .....	8
4.7. Stretched Comparisons to TB .....	9
4.8. Time Resolution of Signal Digitization .....	9
4.9. Depth of Digitization .....	9
4.10. Pulse Separation.....	9
4.11. ADC Range.....	9
4.12. ADC Gain .....	10
4.13. Single-Channel Dead-time.....	10
4.14. Digitizer Module Dead-time .....	10
Requirements and Justification for the Nestor Trigger Board (TB) .....	12
1. Physical Characteristics .....	12
2. Connections.....	12
2.1. Backplane Communications Bus (BCB).....	12
2.2. Nestor Digitizer Modules (NDMs) .....	12
2.3. Slow Controls System (SCS) .....	12

2.4.	Floor-to-Floor Daisy-Chain (FFD) .....	12
3.	Control Functionality .....	12
3.1.	Global Clock .....	12
3.2.	Local Oscillator.....	12
3.3.	Clock Source Selection .....	13
3.4.	Monitoring .....	13
3.5.	Notification .....	13
3.6.	Global Clock to NDM.....	13
3.7.	Master/Slave Selection.....	13
3.8.	Synchronization Trigger Generation.....	13
3.9.	Event Data Gathering and Formatting .....	13
3.10.	Scaler Data Gathering and Formatting .....	13
3.11.	Storage .....	14
3.12.	Data Transfer to Controller.....	14
4.	Trigger Circuitry .....	14
4.1.	Stretched Comparator Signals from NDMs.....	14
4.2.	“Busy” Signal from NDMs.....	14
4.3.	“Busy” Logic .....	14
4.4.	Multiplicity Logic (MLU).....	14
4.5.	Multiplicity Scalers.....	14
4.6.	Global Trigger Decision .....	15
4.7.	Global Trigger Word to NDMs.....	15
4.8.	Time Stamp Counter .....	15
4.9.	Trigger Board Dead-time.....	15
	Requirements and Justification for the Nestor Fiber Communications System (FCS) ....	16
1.	Physical Characteristics .....	16
1.1.	Sphere Station .....	16
1.2.	Shore Station.....	16
2.	Connections.....	16
2.1.	Single-Mode Fibers.....	16
2.2.	Backplane Communications Bus (BCB) .....	16
2.3.	Ethernet .....	16
3.	Functionality .....	17
3.1.	Clock Generation .....	17
3.2.	Clock Distribution.....	17
3.3.	Control Distribution .....	17
3.4.	Data Transfer .....	17
3.5.	GPS Time Stamp.....	17
3.6.	Junction Box Monitor .....	17
3.7.	Data Rates .....	17
	Requirements and Justification for the Nestor Slow Controls System (SCS) .....	19
1.	Physical Characteristics .....	19
2.	Connections.....	19
2.1.	Backplane Communications Bus (BCB).....	19
2.2.	Monitor Inputs .....	19
2.3.	Power On/Off Switch to PMTs.....	19

2.4.	PMT HV Control .....	19
2.5.	Power On/Off Switch to Calibration LEDs .....	19
2.6.	“Fire” Command to Calibration LEDs .....	19
2.7.	“LED Trigger” Command to TB .....	19
3.	Functionality .....	20
3.1.	Digitization of Monitor Signals .....	20
3.2.	PMT Power Switch Control.....	20
3.3.	PMT HV Control .....	20
3.4.	LED Power Switch Control .....	20
3.5.	Remote LED Fire Control.....	20
3.6.	Free-Running LED Fire Control.....	20
3.7.	“LED Trigger” Command to TB .....	20
	Implementation of the Nestor Digitizer Module (NDM).....	21
1.	Introduction.....	21
2.	Clock Distribution.....	22
2.1.	Flash ADC Sampling Clock .....	22
2.2.	FIFO and Comparator Clocks.....	22
3.	Input Channel.....	22
3.1.	Analog Circuit.....	22
3.2.	Digital Logic .....	23
3.2.1.	FIFO.....	23
3.2.2.	Comparator .....	23
3.2.3.	Scaler.....	23
3.2.4.	Stretcher .....	23
3.2.5.	Test Pulse Generator .....	23
4.	Time Stamp Counter .....	24
5.	Memory .....	24
6.	Dead-time.....	25
7.	Operation Control .....	25
	Implementation of the Nestor Trigger Board.....	26
1.	Introduction.....	26
2.	Global Clock .....	27
3.	FIFO.....	27
4.	Trigger Generator.....	27
4.1.	Multiplicity Logic Unit (MLU) .....	27
4.2.	Multiplicity Scalers.....	28
4.3.	Global Trigger.....	28
4.4.	Busy Logic .....	28
5.	Time Stamp Counter .....	29
6.	Memory .....	29
7.	Dead-time.....	30
8.	Operation Control .....	30
	Glossary .....	31

## Nestor Sphere Block Diagram



E.G. Judd, 27th Sept. 2004

## **Requirements and Justification for the Nestor Digitizer Module (NDM)**

### **1. Physical Characteristics**

#### **1.1. Form Factor**

**Requirement:** The NDM must fit into a standard commercial electronics card cage with standard communications backplanes. The card cage must be small enough to be supported inside a 90cm inside-diameter sphere.

**Justification:** Using industry standard form factors and backplanes will simplify the development and production of this module and increase the flexibility of the system by making it possible to add other industry-standard modules. A 90cm inside-diameter sphere is the housing for all electronics for one floor of a Nestor tower.

#### **1.2. Temperature**

**Requirement:** The NDM must be able to operate at temperatures between 14 and 35 degrees Centigrade.

**Justification:** The ocean temperature at the Nestor site is 14 C. During Nestor's March/April tests the temperature inside the sphere leveled out at 29C, and 35C gives us some headroom (see page 25 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)).

#### **1.3. Atmosphere**

**Requirement:** The NDM must operate in a nitrogen atmosphere at atmospheric pressure.

**Justification:** The Nestor titanium spheres are filled with Nitrogen at 1 atmosphere pressure.

#### **1.4. Power**

**Requirement:** The NDM may consume a fraction (currently undefined) of 50W.

**Justification:** During the March/April 2003 tests the total power consumption of the floor was 120W which is nicely within the limits of what can be provided by the shore cable. Each of the 12 optical modules consumed 6W (E.G. Anassontzis et al. (The Nestor Collaboration), NIM-A 479 (2002) 439-455) leaving approximately 50W for the Titanium sphere electronics. With 50W power dissipation during the March/April 2003 tests the temperature inside the Titanium sphere leveled off at 29C, which is a reasonable operating temperature for the electronics. We should therefore make sure that this new electronics falls within that same power budget.

### **2. Connections**

#### **2.1. Backplane Communications Bus (BCB)**

**Requirement:** The NDM must connect to the BCB.

**Justification:** This is the path by which the NDM will communicate with the Fiber Communications System (FCS), which is the system responsible for controlling the operation of this module and reading out its data. The BCB will also provide the NDM with its power and ground connections.

## **2.2. Trigger Board (TB)**

**Requirement:** The NDM must connect to the TB via the Trigger Communications Bus (TCB)

**Justification:** The TB will make the global trigger decision based in part on information it receives from multiple NDMs (see the Trigger Board Requirements section 4). It will then distribute trigger commands, and the global clock, to those NDMs.

## **2.3. PMT signals**

**Requirement:** The NDM must connect to the signal output of 6 PMTs.

**Justification:** This purpose of this module is to digitize those signals.

# **3. Control Functionality**

## **3.1. Global Clock from TB**

**Requirements:** The NDM must accept a global clock provided by the TB, via the TCB, and use it to control the operation of the module.

**Justification:** In order for an NDM to communicate properly with the TB it needs to be operating on exactly the same clock frequency. If the TB sends its clock to the NDM then that condition will be met.

## **3.2. Local Oscillator**

**Requirements:** The NDM must contain a local oscillator that runs at approximately the same frequency as the global clock.

**Justification:** The oscillator will be the clock source that will operate the NDM when it is undergoing early testing in a stand-alone mode.

## **3.3. Clock Source Selection Jumper**

**Requirement:** A jumper will be provided to make a hardware selection between the local oscillator and the external global clock.

**Justification:** The local oscillator is designed purely for a stand-alone test phase. During normal operation the NDM *must* receive its clock from the TB in order to also receive trigger commands. There is no need for a remotely controllable software switch to select between the two clock sources; a hardware switch will be sufficient.

## **3.4. Trigger Word from TB**

**Requirement:** The NDM must also accept a trigger command word from the TB and use it to initiate the processing of this event: fire test pulses if necessary, digitize and store the PMT signals, save the time stamp for this event, format and store all the data, etc....

**Justification:** This is how triggered events will be digitized and stored.

## **3.5. Logic Control**

**Requirement:** The NDM must provide the user with remote control (via the FCS and the BCB) of the comparator thresholds, the length of the stretched comparator signals, the rate at which the scalers are read out and any other aspects of the digitization circuitry that might need to be adjusted.

**Justification:** The user needs control these aspects of the NDM's operation.

**3.6. Time Stamp Counter**

**Requirement:** The NDM must count global clock cycles starting from either power-on or a user initiated reset. When a trigger is received the current value of this counter must be included in the data stream.

**Justification:** This will enable the user to study the time distribution of events. It will also make it possible to cross-calibrate the relative time between Nestor floors to an integral number of global clock ticks. This calibration will be performed by two methods. The first technique will use the synchronization trigger distributed with a known time offset to all floors via the Floor-to-Floor Daisy-Chain (FFD). The second technique will use data from LED events, when an LED in a known location is fired with a known duration and amplitude.

**3.7. Event Data Gathering and Formatting**

**Requirement:** Once the PMT signals have been digitized their data and any other relevant information (the time stamp, the trigger word, etc...) must be gathered into a formatted event-data package.

**Justification:** The data needs to be gathered into a known format in order for the user to analyze it.

**3.8. Scaler Data Gathering and Formatting**

**Requirement:** Periodically, at a user-settable rate, the current values of all scalers and any other relevant information (e.g. the time stamp) must be gathered into a formatted scaler-data package.

**Justification:** The data needs to be gathered into a known format in order for the user to analyze it.

**3.9. Storage**

**Requirement:** The NDM needs at least 463 Mbytes of memory to store up to 1.5 hours worth of scaler and event data internally before they are read-out by the FCS.

**Justification:** Data will be read out periodically via the BCB and the FCS. This module needs to store events until they are read out. (See “Implementation of the Nestor Digitizer Module (NDM)”, section 5 for the calculation of the event size and data rate).

**3.10. “Busy” Signal to TB**

**Requirement:** The NDM must inform the TB when it is unable to receive new triggers, either because it is busy processing a previous trigger or because the memory is full so there is nowhere to store new data. This information will be sent via the TCB.

**Justification:** There is no point in the TB issuing a trigger if an NDM is unable to process the command.

**3.11. Data Transfer to FCS**

**Requirement:** Data must be sent to the FCS on request via the BCB.

**Justification:** The FCS is the system responsible for transmitting the data to shore.

## 4. Digitization Circuitry

### 4.1. Ground and Power Separation

**Requirement:** The analog and digital power and ground circuits must all be separated from each other. The two ground circuits should meet only at the connection to the BCB ground pin. The two power circuits should meet only at the connection to the BCB power pin.

**Justification:** This will prevent digital noise being picked up by the analog circuits.

### 4.2. Termination of PMT signals

**Requirement:** Each of the 6 channels must be correctly terminated into 50Ω

**Justification:** If this is not done correctly the PMT signals will be distorted

### 4.3. Noise

**Requirement:** The noise generated in the PMT signals by the following analog circuitry must be less than 10 mV which is less than 10% of a 120 mV single photoelectron peak (see [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html) on page 36) and is down in the region already dominated by the PMT dark current.

**Justification:** It must still be possible to distinguish single photoelectron signals from the noise.

### 4.4. Comparison of PMT Signals to Threshold

**Requirement:** Each of the 6 PMT signals must be compared to a threshold which must be useable down to  $\frac{1}{4}$  of a photoelectron, which is 30mV for Nestor. The threshold must also be remotely adjustable, on a channel-by-channel basis.

**Justification:** If we discriminate reliably at  $\frac{1}{4}$  of a photoelectron then our ability to detect single photoelectrons approaches 100%. The control is necessary for fine adjustments (if the PMTs are not properly gain-matched then they will produce single photoelectron pulses of different sizes) and in case we need to raise the comparison threshold on a noisy PMT.

### 4.5. Scalers for Comparator Output

**Requirement:** A scaler is needed for each of the 6 channels to count the number of hits on that channel. A hit occurs when the PMT signal goes above threshold and the comparator output changes from 0 to 1. Each scaler should be reset to zero after it has been read out.

**Justification:** It is necessary to monitor the hit rate on each channel individually in order to monitor their performance.

### 4.6. Stretching of Comparator Outputs

**Requirement:** The comparator output pulses must be stretched to be up to 160ns long. The actual length must be remotely adjustable. One adjustment for all 6 channels is sufficient; there is no need to adjust the length for each channel individually.

**Justification:** 160ns is the maximum separation between the earliest and latest pulses from a single muon. It is set by the time it takes light to traverse a 32m diameter floor, traveling at  $\frac{3}{4}$  of the speed of light in a vacuum (143ns), plus the 7ns transit time spread (worst case, single p.e.) of the two PMTs. The

minimum stretched pulse length is equal to the period of the global clock that will be used by the TB to implement the trigger logic. Any pulse that is shorter than this would be missed by that trigger logic (see “Requirements and Justification for the Nestor Trigger Board (TB)”, sections 3.1 and 4.4).

Stretching each comparator output pulse will make it possible to trigger on coincidences between early and late signals from one muon.

**4.7. Stretched Comparisons to TB**

**Requirement:** The stretched comparator outputs must be sent to the TB via the TCB.

**Justification:** The TB uses these signals to make its global trigger decision.

**4.8. Time Resolution of Signal Digitization**

**Requirement:** Once a trigger command has been received information about the PMT pulses must be digitized and recorded in such a way that the NDM contributes no more than 0.5ns to the final timing resolution of the leading edge time in Nestor.

**Justification:** The final timing resolution is defined as

$$\sigma = \sqrt{\sigma(PMT)^2 + \sigma(el)^2}$$
 where  $\sigma(PMT)$  is the timing resolution of the PMTs and  $\sigma(el)$  is the timing resolution of the NDM.  $\sigma(el)$  should be no more than 50% of the final timing resolution,  $\sigma$ . In Nestor the best value of  $\sigma(PMT)$  is set by the Transit Time Spread (TTS) of multi-photoelectron pulses, which is 2ns. A TTS of 2ns corresponds to  $\sigma(PMT)$  of 0.85ns, which implies  $\sigma(el)$  must be no more than 0.5ns.

**4.9. Depth of Digitization**

**Requirements:** The total time window must be at least 470ns.

**Justification:** This is set by twice the maximum separation between the earliest and latest pulses. The maximum pulse separation is 235ns, given by the sum of 160ns (floor transit time, see section 4.6) and 75ns to allow for the late arrival of slow shower particles. If the trigger is based on the earliest pulse then it will be necessary to digitize and store the 235ns after the trigger. However, if the TB waits until the latest pulse arrives before issuing a trigger then it will be necessary to digitize and store the 235ns preceding the trigger. A time window of at least 470ns should cover both possibilities.

**4.10. Pulse Separation**

**Requirements:** It is necessary measure the time profile of each PMT signal in order to separate out pulses that are so close they overlap.

**Justification:** Overlapping pulses are produced when the PMT signals are the result of a shower of particles; the prompt muon and a cascade of slower particles. In order to reconstruct the track of the prompt muon it is necessary to separate out its pulses from those of the slower cascade particles.

**4.11. ADC Range**

**Requirements:** The ADC system used to measure the time profile (see 4.10 above) needs to be able to deal with pulses ranging in amplitude from zero up to 8V.

**Justification:** 8V is where the output of the PMT saturates so no pulse will be larger than that.

#### 4.12. ADC Gain

**Requirements:** The gain does not need to be constant over the full 8V range. A high gain region is needed for pulses up to around 2V. For the larger pulses a lower gain would be adequate. A non-linear ADC, or multiple ADCs running in parallel with different gains, would meet the requirement.

**Justification:** A typical single-photoelectron pulse is around 120mV. However, these pulses can be as small as 70mV or as large as 200mV with a tail stretching up to higher values. A typical double-photoelectron pulse is around 260mV but they can be as small as 200mV or as large as 360mV again with a large high tail (see page 36 of

[http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)). During the majority of multiplicity-triggered events each PMT typically sees up to 4 photoelectrons from the prompt muon. By extrapolation from the single and double-photoelectron pulse sizes, during a 4-photoelectron event a PMT could easily see pulses up to 2V. Over the size range of these prompt pulses it is necessary to have a high gain in order to make the most accurate measurement of the time profile of the leading edge of each pulse. This is necessary to maximize the efficiency of reconstruction of the prompt muon's trajectory. By contrast, the very large pulses, where the PMT output is starting to saturate, are produced when a PMT is hit by multiple photons from multiple slow shower particles. Since it is not possible to distinguish individual slow shower particles it is not necessary to measure the time profile of their pulse train with the same level of accuracy.

#### 4.13. Single-Channel Dead-time

**Requirement:** Each channel must be dead for no more than 1  $\mu$ s after each pulse.

**Justification:** During the Nestor tests in March/April 2003 the maximum single PMT rates seen were bursts of 400 kHz on the upward-looking PMTs. Typical rates on these PMTs were more like 70 kHz (see page 27 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)). The downward-looking PMTs saw typical rates of 50 kHz with bursts up to 80 kHz (see page 28 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)). If each channel is dead for 1  $\mu$ s after its input signal goes over the comparison threshold then the upward-looking channels will typically be dead for 7% of the time, and the downward-looking channels will typically be dead for 5% of the time. This is acceptable.

#### 4.14. Digitizer Module Dead-time

**Requirement:** The NDM must be dead for no more than 10 ms after each trigger issued by the TB. This includes the time to format the data and store it.

**Justification:** During the tests in March/April 2003 (using a 12m diameter floor) the trigger rates were around 3 Hz when a 4-fold coincidence was required with the comparison thresholds set to  $\frac{1}{4}$  of a photoelectron (see pages 31 and 39 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)). However, during the commissioning and debugging process triggers that used a lower coincidence level were used, leading to higher trigger rates. During normal operation with a full-size floor (32m diameter) the comparator outputs will be stretched for longer to cover the full 32m transit time so the random

coincidence rate from Potassium-40 background is expected to rise to 30 Hz. If the module goes dead for 10 ms then the maximum steady state trigger rate than can be achieved is 100Hz which is acceptable. It should be noted that the actual implementation we are planning will allow a steady state trigger rate of 27 kHz (see “Implementation of the Nestor Digitizer Module (NDM)”, section 6).

## **Requirements and Justification for the Nestor Trigger Board (TB)**

### **1. Physical Characteristics**

See “Requirements and Justification for the Nestor Digitizer Module (NDM)”, section 1.

### **2. Connections**

#### **2.1. Backplane Communications Bus (BCB)**

**Requirement:** See “Requirements and Justification for the Nestor Digitizer Module (NDM)”, section 2.1.

#### **2.2. Nestor Digitizer Modules (NDMs)**

**Requirement:** The TB must connect to two NDMs using the TCB.

**Justification:** The TB must receive comparator signals from both NDMs for use in making the trigger decision. Once a trigger decision has been made that information must be passed to the NDMs to initiate digitization and read-out of the PMT data.

#### **2.3. Slow Controls System (SCS)**

**Requirement:** The TB must connect to the “LED Trigger” output connector of the SCS.

**Justification:** The SCS needs to use this connection to force this module to issue a trigger to the NDMs with a known, fixed, start time, when an LED event is initiated.

#### **2.4. Floor-to-Floor Daisy-Chain (FFD)**

**Requirement:** The TB must connect to the TBs in the other floors via the Floor-to-Floor Daisy Chain (FFD).

**Justification:** The FFD can be used by any TB to distribute a signal to all other floors. This communications path will be used as part of the procedure to synchronize data between multiple floors.

### **3. Control Functionality**

#### **3.1. Global Clock**

**Requirements:** The TB must accept an externally provided global clock from the FCS and use it to control the operation of the module.

**Justification:** The task of combining data from multiple Nestor floors will be much easier if all the electronics is operating using a common clock frequency. A global clock will be provided to all Nestor floors via the FCS.

#### **3.2. Local Oscillator**

**Requirements:** The TB must contain a local oscillator that runs at approximately the same frequency as the global clock.

**Justification:** The oscillator will be the default clock source that the TB will use until it is instructed to switch to the global clock. It will also be the backup clock source should that external global clock fail.

**3.3. Clock Source Selection**

**Requirement:** In response to user commands the TB must select either the local oscillator or the externally provided global clock.

**Justification:** The user needs to be able to select which clock is used as the system clock by the TB.

**3.4. Monitoring**

**Requirement:** The TB must monitor the global clock and switch back to the local oscillator if the global clock fails.

**Justification:** If the global clock fails then the TB will stop working unless it is provided with an alternative clock.

**3.5. Notification**

**Requirement:** In response to a request from the user the TB must generate a data word that describes the current clock status: “using global clock”, “global clock failed so switched back to local oscillator”, etc....

**Justification:** The user will need to know which clock is being used.

**3.6. Global Clock to NDM**

**Requirement:** The TB must distribute whichever clock is currently selected to the two NDMs over the TCB.

**Justification:** The NDMs need this clock to operate.

**3.7. Master/Slave Selection**

**Requirement:** The user must be able to designate this TB as either the master of the FFD or a slave.

**Justification:** Only one TB can drive a synchronization signal on the FFD at any one time; all others TB's must receive and process that signal. In order to cross-check the synchronization between floors each floor must be selected as the master in turn.

**3.8. Synchronization Trigger Generation**

**Requirement:** If this TB has been designated as the FFD-Master then, at a user-settable time, it must generate a synchronization trigger, process it internally (see Section 4.6 below) and drive a pulse to all the TBs on the other floors via the FFD.

**Justification:** This will guarantee that each slave floor receives the synchronization trigger with a fixed time offset with respect to the master floor.

**3.9. Event Data Gathering and Formatting**

**Requirement:** Whenever a trigger decision is made the TB must assemble all of its data (the 12 comparator signals from the NDMs, any information from the SCS, the time stamp, etc...) into a formatted data package.

**Justification:** The data needs to be gathered into a known format in order for the user to analyze it.

**3.10. Scaler Data Gathering and Formatting**

**Requirement:** Periodically, at a user-settable rate, the current values of all scalers and any other relevant information (e.g. the time stamp) must be gathered into a formatted scaler-data package.

**Justification:** The data needs to be gathered into a known format in order for the user to analyze it.

### 3.11. Storage

**Requirement:** The TB needs at least 10 Mbytes of memory to store up to 1.5 hours of scaler and event data internally before they are read-out by the FCS.

**Justification:** Data will be read out periodically via the FCS and the BCB.

The TB needs to store events until they are read out (See “Implementation of the Nestor Trigger Board (TB)”, section 6 for the calculation of the event size and data rate).

### 3.12. Data Transfer to Controller

**Requirement:** Data must be sent to the FCS on request via the BCB.

**Justification:** The FCS is responsible for transmitting the data to shore.

## 4. Trigger Circuitry

### 4.1. Stretched Comparator Signals from NDMs

**Requirement:** The TB must accept and store the 6 stretched comparator signals that it receives from each of the 2 NDMs via the TCB.

**Justification:** This is the information that this TB will use to make its trigger decision.

### 4.2. “Busy” Signal from NDMs

**Requirement:** The TB must accept a busy signal from each of the two NDMs via the TCB.

**Justification:** This is how the NDMs will tell the TB when they are unable to process new trigger commands.

### 4.3. “Busy” Logic

**Requirement:** The TB must disable the trigger logic if either of the two NDMs is busy, or if its internal memory is full so there is no space to store data from a new trigger.

**Justification:** There is no point in issuing a trigger if it cannot be processed and stored.

### 4.4. Multiplicity Logic (MLU)

**Requirement:** The 12 stretched comparator signals from the two NDMs must all go to a configurable multiplicity logic unit (MLU) that will calculate the multiplicity by counting how many of the 12 signals are on. The multiplicity is also known as the “coincidence level”. The MLU must then produce a trigger signal whenever the multiplicity exceeds a user-settable threshold. It must be possible for the user to set up multiple trigger conditions, with independent thresholds and independent prescales running in parallel. A trigger would be issued whenever any prescale is satisfied.

**Justification:** This is how the sampling and storage of the PMT signals are initiated. The capability of using multiple triggers in parallel is needed in order to make sure rare, important events are triggered while simultaneously continuing to trigger on a subset of very common events.

### 4.5. Multiplicity Scalers

**Requirement:** The multiplicity (i.e. the coincidence level) can have one of 13 values between 0 and 12. 13 scalers need to be implemented. Every tick of the clock the scaler corresponding to the current multiplicity value should be

incremented by 1. All 13 scalars should be reset to zero after they have been read out.

**Justification:** It is necessary to monitor the rate at which each coincidence level is occurring.

**4.6. Global Trigger Decision**

**Requirement:** The MLU trigger must be combined with any LED trigger (received from the SCS) and any synchronization trigger (either generated here or received via the FFD) to create a global trigger word that will initiate readout of the data in both this TB and the two NDMs.

**Justification:** This is how the final trigger decision is made in this module.

**4.7. Global Trigger Word to NDMs**

**Requirement:** The global trigger word must be sent to the two NDMs via the TCB.

**Justification:** This is how the NDMs will be told to digitize and store their data.

**4.8. Time Stamp Counter**

**Requirement:** The TB must count global clock cycles starting from either power-on or a user initiated reset. When a trigger is generated the current value of this counter must be included in the data stream.

**Justification:** This will enable the user to study the time distribution of events. It will also make it possible to cross-calibrate the relative time between Nestor floors to an integral number of global clock ticks. This calibration will be performed by two methods. The first technique will use the synchronization trigger distributed with a known time offset to all floors via the Floor-to-Floor Daisy-Chain (FFD). The second technique will use data from LED events, when an LED in a known location is fired with a known duration and amplitude.

**4.9. Trigger Board Dead-time**

**Requirement:** The TB must be dead for no more than 10 ms after each trigger is issued. This includes the time to format the data and store it.

**Justification:** During the tests in March/April 2003 (using a 12m diameter floor) the trigger rates were around 3 Hz when a 4-fold coincidence was required with the comparator thresholds set to  $\frac{1}{4}$  of a photoelectron (see pages 31 and 39 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html)). However, during the commissioning and debugging process triggers that used a lower coincidence level were used, leading to higher trigger rates. During normal operation with a full-size floor (32m diameter) the comparator outputs will be stretched for longer to cover the full 32m transit time so the random coincidence rate from Potassium-40 background is expected to rise to 30 Hz. If the module goes dead for 10 ms then the maximum steady state trigger rate than can be achieved is 100Hz. It should be noted that the actual implementation we are planning will allow a steady state trigger rate of 1.4 MHz (see "Implementation of the Nestor Trigger Board (TB)", section 7) if the TB is operating in stand-alone mode with no NDMs to throttle the rate.

## **Requirements and Justification for the Nestor Fiber Communications System (FCS)**

### **1. Physical Characteristics**

#### **1.1. Sphere Station**

**Requirement:** See “Requirements and Justification for the Nestor Digitizer Module (NDM)”, section 1.

#### **1.2. Shore Station**

**Requirement:** The shore station of the FCS must be a commercial computer running an industry-standard operating system.

**Justification:** Using commercial hardware and operating systems will simplify the development of this system and increase the flexibility by making it possible to easily upgrade and expand it.

### **2. Connections**

#### **2.1. Single-Mode Fibers**

**Requirement:** The FCS must be set up as a “star-topology” fiber optic network. The Shore Station will be the hub, connecting to the shore end of all the 35km-long single-mode fibers in the shore-tower cable. The Sphere Stations will be the nodes, with one Sphere Station in the Titanium electronics sphere on each Nestor floor connecting to the tower end of one single-mode fiber. The FCS must provide bi-directional communication (i.e. both transmit and receive) and global-clock distribution on every fiber.

**Justification:** The 35km shore-tower cable contains 15 working single-mode fibers. Ultimately there will be at least 8 floors in the tower so there is only one fiber available per floor (see page 12 of [http://www.nestor.org.gr/hena\\_paris/index.html](http://www.nestor.org.gr/hena_paris/index.html) ). That one fiber must be used to bring in a global clock and control information from the shore, and then send data back to the shore. The global clock will originate from one source on shore and will then be fanned out and distributed over all 12 fibers.

#### **2.2. Backplane Communications Bus (BCB)**

**Requirement:** The FCS Sphere Station must connect to the BCB in its sphere.

**Justification:** Control information for the NDMs, TB and SCS will come from shore over the fiber, and digitized data will be read-out over the same fiber. The BCB will be used to distribute those commands, and transfer data into and out of those subsystems. The FCS provides the interface between the fiber and the BCB.

#### **2.3. Ethernet**

**Requirement:** The FCS Shore Station must have a high-speed Ethernet connection.

**Justification:** This will enable the operator to control and monitor the system remotely, and also make electronic distribution of the data possible.

### 3. Functionality

#### 3.1. Clock Generation

**Requirement:** The FCS Shore Station must generate a global clock signal with a frequency in the range 50-70MHz (this is frequency range of the word-rate clock used to keep both ends of a fiber optic link synchronized).

**Justification:** This is the global clock that will be distributed to all TBs and all NDMs in all Nestor spheres.

#### 3.2. Clock Distribution

**Requirement:** The FCS Shore Station must distribute this global clock on all the fibers to all the spheres.

**Justification:** This will ensure that the TBs and all the NDMs in all floors are using the same frequency global clock, which will make cross-calibrations between the floors much easier.

#### 3.3. Control Distribution

**Requirement:** The FCS Shore Station must decode the address information of incoming communications from the operator and route the communication to the correct fiber for the sphere that is being addressed. At the other end of that fiber the FCS Sphere Station must provide an interface to the BCB to enable the incoming communication to reach the module (TB, NDM, SCS) it is intended for.

**Justification:** The user will need to be able to communicate with each of the other modules in the sphere individually in order to control their operation.

#### 3.4. Data Transfer

**Requirement:** Under shore control the FCS Sphere Station must read the digital data that is stored in the other modules in the sphere and transmit that data to shore over the fiber.

**Justification:** This is how the data gets to shore. A “Pull Architecture” under shore control simplifies the procedure and removes any need to deal with collisions that would result if multiple sphere systems were trying to push data over one fiber.

#### 3.5. GPS Time Stamp

**Requirement:** Whenever the FCS Shore Station reads data from a sphere the data packet must be tagged with the current GPS time.

**Justification:** This will make it possible to associate events in Nestor with events in the outside world, e.g. Gamma Ray Bursts.

#### 3.6. Junction Box Monitor

**Requirement:** Whenever the FCS Shore Station reads data from a sphere it must also read and save the Junction Box monitor information.

**Justification:** It is necessary to monitor the health of the Junction Box.

#### 3.7. Data Rates

**Requirement:** The FCS must be capable of transferring data at rates of at least 0.2 Mbytes/second on any one fiber.

**Justification:** Each event will consist of approximately 6 kilobytes; 2.85 Kbytes from each of 2 NDMs and 54 bytes from the TB (see Section 5 in the Digitizer Module Implementation document and section 6 in the Trigger

Board Implementation document). At the expected event rate of 30 Hz this corresponds to a data rate of 0.2 Mbytes/second. Since fiber optic links are typically capable of data rates in excess of 100 Mbytes/second this should not be a problem.

## **Requirements and Justification for the Nestor Slow Controls System (SCS)**

### **1. Physical Characteristics**

See “Requirements and Justification for the Nestor Digitizer Module (NDM)”, section 1.

### **2. Connections**

#### **2.1. Backplane Communications Bus (BCB)**

**Requirement:** See “Requirements and Justification for the Nestor Digitizer Module (NDM)”, section 2.1.

#### **2.2. Monitor Inputs**

**Requirement:** The SCS must receive signals from various monitoring devices: PMT HV monitor, acoustic position monitor, power supply voltage, compass, accelerometer, humidity and temperature sensors, etc.... At least some of these connections will be of RS232 type. The full list is currently undefined.

**Justification:** These conditions need to be monitored. Nestor already has a variety of monitoring devices that use the RS232 interface and these must be accommodated.

#### **2.3. Power On/Off Switch to PMTs**

**Requirement:** The SCS must connect to the Power on/off switch for each PMT.

**Justification:** The PMTs need to be switched on in order to function, and off if there is a problem (e.g. if a Benthos sphere leaks and shorts all its electronics).

#### **2.4. PMT HV Control**

**Requirement:** The SCS must connect to the low voltage control input to each PMT.

**Justification:** This signal is used to control the Cockcroft-Walton base and set the required HV

#### **2.5. Power On/Off Switch to Calibration LEDs**

**Requirement:** The SCS must connect to the Power on/off switch of each LED.

**Justification:** The LEDs must be switched on in order to function, and off if there is a problem (e.g. if the housing leaks and shorts all its electronics).

#### **2.6. “Fire” Command to Calibration LEDs**

**Requirement:** The SCS must connect to the LED control input.

**Justification:** This is used to actually fire the LEDs.

#### **2.7. “LED Trigger” Command to TB**

**Requirement:** The SCS must connect to the TB’s “LED Trigger” input.

**Justification:** The SCS needs to tell the TB when it has fired the LEDs.

### 3. Functionality

#### 3.1. Digitization of Monitor Signals

**Requirements:** The SCS must provide the ability for the user to control the digitization of the analog monitor inputs (e.g. set ADC gate width, initiate digitization, etc...) and read-out the digitized data via the BCB and the FCS.

**Justification:** Only digitized data can be sent to shore via the FCS, and the monitor data needs to be saved and analyzed along with the PMT data.

#### 3.2. PMT Power Switch Control

**Requirement:** The SCS must provide the ability for the user to switch on and off the power to individual PMTs.

**Justification:** The user needs this basic control over individual PMTs.

#### 3.3. PMT HV Control

**Requirement:** The SCS must provide the ability for the user to set the HV on individual PMTs.

**Justification:** The user needs control over individual PMT's HV.

#### 3.4. LED Power Switch Control

**Requirement:** The SCS must provide the ability for the user to switch on and off individual LEDs.

**Justification:** The user needs this basic control over which LED's are available to fire and which are switched off.

#### 3.5. Remote LED Fire Control

**Requirement:** The SCS must provide the ability for the user to fire a specified LED, and control the light intensity of the pulse. The control circuitry must be stable against small changes in electrical components (this was a problem in the past) and must produce very stable, sharp pulses (1-2ns).

**Justification:** The user needs to be able to initiate a specific calibration event.

#### 3.6. Free-Running LED Fire Control

**Requirement:** The SCS must provide the ability for the user to set up an automatic sequence of LED firing events specifying the frequency (up to at least 100Hz), LED selection, and light intensity of each event. These pulses have the same requirements of sharpness and stability as before.

**Justification:** This would be useful to ensure a regular sequence of calibration events interspersed throughout the data.

#### 3.7. "LED Trigger" Command to TB

**Requirement:** The SCS must generate a trigger signal for the TB whenever any LED is fired.

**Justification:** In order to use the LED system to fully investigate the response characteristics of the system (e.g. effects of light amplitude, cables, differences between PMTs, differences due to different levels of multiplicity logic, etc...) it would be helpful for the LED events to always be triggered with a fixed time relationship to the actual LED firing.

# Implementation of the Nestor Digitizer Module (NDM)

## 1. Introduction

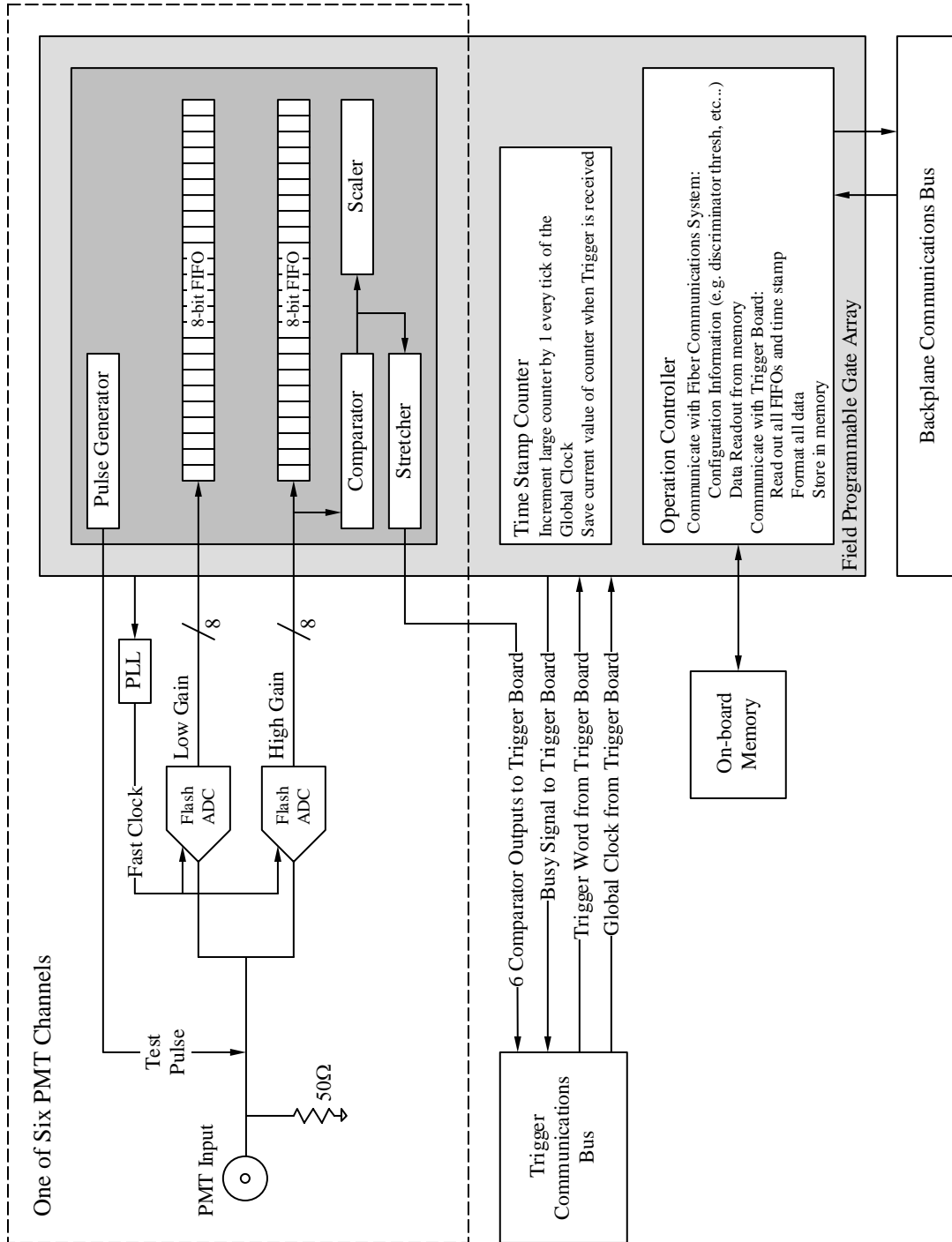


Figure 1: A schematic Diagram of the Nestor Digitizer Module (NDM)

The Nestor Digitizer Module (NDM) is a 6-channel module that uses flash ADC chips to record the waveforms of each input pulse. Each NDM is designed to operate as one of a pair that communicates with a TB for triggering purposes. A schematic diagram of the module is shown in Fig. 1. The NDM consists of 6 identical input channels designed to be driven by a PMT output, a Time Stamp Counter, a memory block and control logic that provides an interface to an external CPU. All the logic is implemented in a Field Programmable Gate Array (FPGA). The various functional blocks will be described in the following sections of this document.

## **2. Clock Distribution**

The NDM will receive an external global clock from the TB (see Section 2 of “Implementation of the Nestor Trigger Board”) via the TCB. This clock operates in the range of 50-70MHz. A jumper-selectable local oscillator will also be provided for stand-alone testing.

### **2.1. Flash ADC Sampling Clock**

Each NDM will then multiply the frequency of this global clock to make its flash ADC sampling clock. The flash ADC sampling clock will have a frequency of 500MHz. The multiplication will be accomplished through a combination of using the Delay Locked Loop (DLL) inside the FPGA and then an external phase locked loop chip (PLL).

### **2.2. FIFO and Comparator Clocks**

The FPGA must also generate its local FIFO and comparator clocks from the global clock. The flash ADC chips include an integrated output demultiplexer that reduces the output data rate to one half of the sampling clock rate, i.e. 250MHz. The FPGA will use its DLL to generate this clock from the global clock.

## **3. Input Channel**

The NDM will implement 6 identical input channels, each of which contains 2 flash ADC chips and logic inside an FPGA.

### **3.1. Analog Circuit**

Each input will be terminated to provide the correct load for the input pulses. Since the expected pulses in Nestor are already large (1 photoelectron produces a 120mV pulse) there will be no further amplification. Instead the pulse will be immediately driven to the inputs of two flash ADC chips. These ADCs will run at 500MHz so every 2ns the waveform will be sampled and its amplitude digitized to an 8-bit ADC value. The two ADCs will operate with different gains; the high-gain ADC will produce a full-scale output for a 2V input signal and the low-gain ADC will not saturate until the input amplitude is 8V. Each ADC then demultiplexes its output so two 8-bit values are produced, in parallel, at a rate of 250MHz. The digital output of the ADCs will then be passed to the FPGA.

### **3.2. Digital Logic**

The digital logic for all 6 input channels will be implemented inside 1 FPGA. Each channel will include 3 sections: two FIFOs, a comparator and a test pulse generator.

#### **3.2.1. FIFO**

The input from each flash ADC chip will be recorded in a 16-bit wide FIFO. Data will shift through the FIFO at the same speed as the flash ADCs output data rate, i.e. 250MHz. The FIFO must be deep enough to hold the data from the 235ns before receiving a trigger from the TB. At a speed of 250MHz, or 4ns per sample, this corresponds to a minimum of 59 samples. When a trigger is received from the TB all 59 samples currently stored in the FIFO will be read out, and then 59 more to record the data from the 235ns after the trigger was generated.

#### **3.2.2. Comparator**

In parallel with the FIFO the output of the high-gain flash ADC will be sent to a comparator circuit to make the signal that goes to the TB. Every tick of the 250MHz clock the current 8-bit ADC value will be compared to a user-settable 8-bit threshold. This could be implemented in a look-up table to save time. The result of the comparison would be a bit stream where “1” indicates the ADC was greater than the threshold and “0” indicates it was not. That bit stream would go to the input of a scaler and a stretcher.

#### **3.2.3. Scaler**

The input to the scaler is the output of the comparator. The scaler will be at least a 20-bit counter that starts from 0 at power-up and can be reset to 0 either by user command or after it has been read. The current value of the counter will be incremented by 1 whenever a transition from “0” to “1” is detected in the input. A 20-bit counter is sufficient to count all hits that occur during a 400kHz burst assuming the scaler is read at least once every 2.5 seconds.

#### **3.2.4. Stretcher**

The input to the stretcher is also the output of the comparator. Whenever a transition from “0” to “1” is detected a pulse would be generated with a length defined in a user-settable register. The length would be between 20ns (5 ticks of the 250 MHz clock, which is 1 tick of the 50 MHz global clock from the TB) and 160ns (40 ticks of the 250 MHz clock). Finally, the status (“0” or “1”) of this stretched pulse would be latched every tick of the slower global clock. That latched status will be sent to the TB, via the TCB, for it to use in making the trigger decision.

#### **3.2.5. Test Pulse Generator**

Logic will be implemented to generate a test pulse that can be fed into the input channel just upstream of the split between the two flash ADC chips. Both the duration and amplitude can be specified by the user.

#### 4. Time Stamp Counter

In order to combine data from different floors it is necessary to know as accurately as possible when the trigger was issued on each floor. This information will be recorded with a time stamp counter. This is a large counter that increments by 1 every tick of the global clock. This global clock frequency is in the range of 50-70MHz (see Section 2), much slower than the rate at which the flash ADC chips are sampling (500MHz). In this frequency range a 51-bit counter will take over a year to count from 0 to its maximum value. Since typical deployments will last for less than a year this counter will allow each event to be uniquely identified. The current value of the counter will be latched and saved in the data stream whenever a trigger from the TB is received. This will narrow down the trigger time to within one tick of that slow global clock. The trigger time will be more tightly constrained within that global clock tick from an analysis of the recorded data.

#### 5. Memory

The NDM will contain enough local memory to store up to 1.5 hours worth of triggered events and scaler data. Table 1 lists the components of each event type and their size, assuming the scalers are read at a 1 Hz rate. The total comes to around 463 Mbytes which can fit into a commercial 512 Mbytes SIMM (DRAM).

	Triggered Events	Scaler Data	Total
Components	<b>Header:</b> 10 bytes <b>Time Stamp:</b> 7 bytes (51 bit counter) <b>Trigger Word:</b> 1 byte <b>FIFO:</b> 2832 bytes (12 Flash ADCs, 118 time samples, 2 bytes/sample)	<b>Header:</b> 10 bytes <b>Time Stamp:</b> 7 bytes (51-bit counter) <b>Scaler:</b> 18 bytes (6 scalers, 3 bytes/scaler)	
Event Size	2.85 Kbytes	35 bytes	
Event Rate	30 Hz	1 Hz	
Memory used in 1.5 hours	462 Mbytes	0.2 Mbytes	~ 463 Mbytes

Table 1: Event Sizes in the NDM

The NDM memory will be controlled by the operation control logic in the FPGA. The rate at which new data is written to the DRAM is approximately 86 Kbytes/second; 2.85 Kbytes/event at a 30 Hz event rate. This is much slower than the DRAM access time, which is approximately 80 Mbytes/second. It will therefore be possible to interleave the read and write operations so they do not interfere with each other. As a result writing new events will not need to be suspended while currently stored events are read out so this memory will have no effect on the dead-time of the NDM.

## **6. Dead-time**

The expected dead-time of the NDM is very short compared to the 10ms requirement (see “Requirements and Justification for the Nestor Digitizer Module (NDM)”, sections 4.13 and 4.14). The flash ADCs run continuously so they will introduce no dead-time into the system. Of the remaining processes involved in storing data from a triggered event (reading the FIFOs, saving the current time stamp and writing all the data into the memory) the memory access is expected to be the limiting time factor. Commercial DRAM can typically be accessed at 80 Mbytes/second (8 bytes at 10 MHz). At this rate it would take 36  $\mu$ s to write one event of 2.85 Kbytes into the memory. The NDM dead-time is therefore expected to be approximately 36  $\mu$ s, leading to a maximum steady-state event rate of 27 kHz.

## **7. Operation Control**

The operation control logic is responsible for configuring the NDM, reading out the data, storing it and communicating with the FCS over the BCB. In response to commands from the FCS it will set the comparator thresholds, the stretcher length, the properties of the test pulses, and any other user-settable feature of this module. When a global trigger word is received from the TB this logic component will take the appropriate action, e.g. gather the data from the 12 ADCs, the time stamp and all the trigger information, form it into one data block and store it in the local, on-board, memory. The operation controller will also send a “busy” signal to the TB to disable the generation of new triggers when the NDM is busy. This can happen in one of 2 circumstances: the system could be busy reading out data from a previous trigger or the memory could be full so there is nowhere to store any new data. Under normal conditions only the first circumstance should actually occur. This signal will be sent via the TCB. At a user-settable rate the operation control logic will also read and clear all the scalers and store that data in the on-board memory. Finally, and again under control of the FCS, it will extract blocks of data from the local memory and pass them to the FCS over the BCB.

## Implementation of the Nestor Trigger Board

### 1. Introduction

The Trigger Board is designed to combine the 12 comparator signals from 2 NDMs, an LED trigger from the SCS and any Synchronization triggers, to make the final trigger decision for this Nestor floor. A schematic diagram of the board is shown in Figure 2.

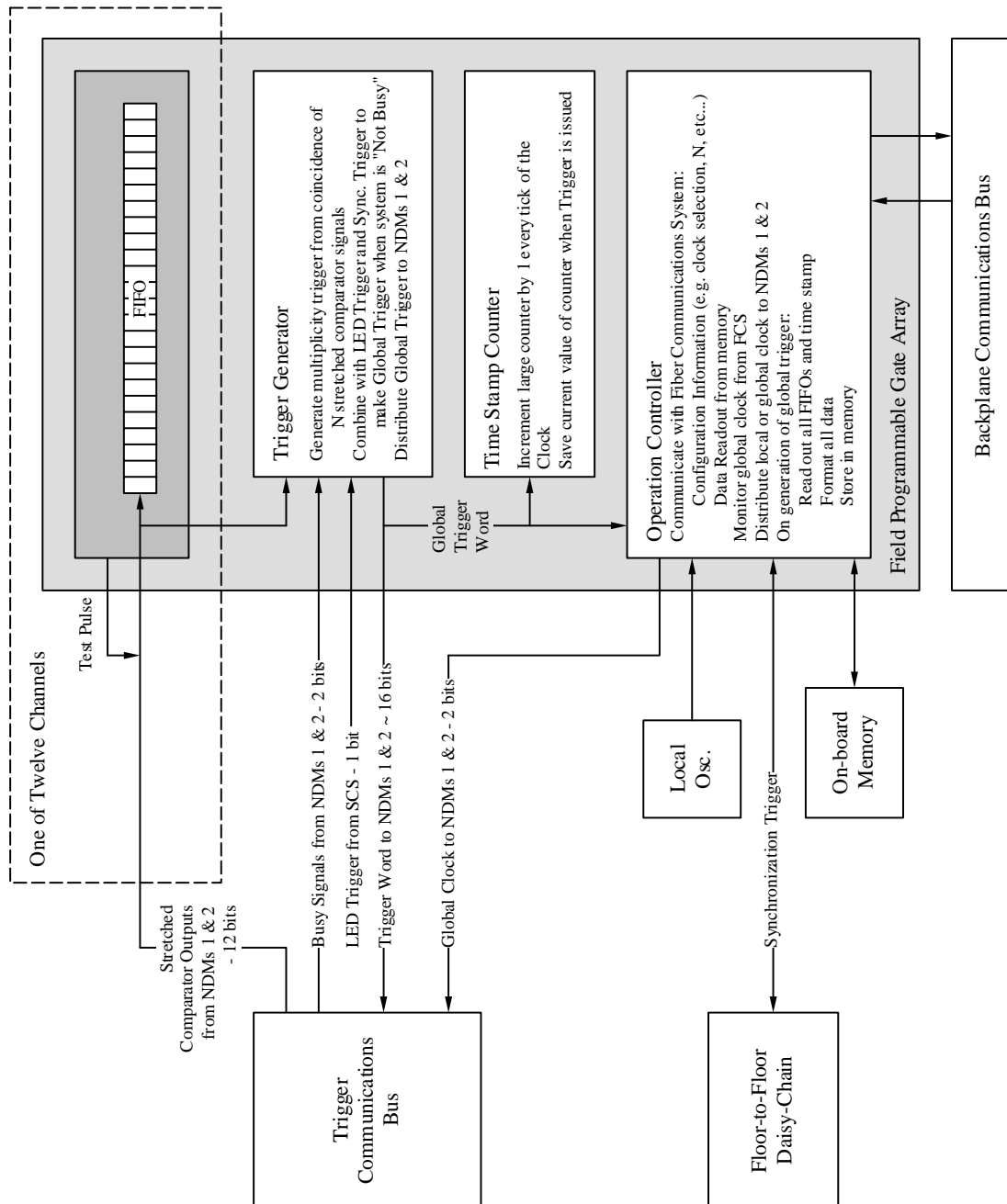


Figure 2: A schematic Diagram of the Nestor Trigger Board (TB)

The incoming comparator data will be saved in FIFOs. The multiplicity trigger decision will be made by a multiplicity logic unit (MLU). The TB is also responsible for selecting and distributing the global clock to the two NDMs. All the logic is implemented in a Field Programmable Gate Array (FPGA). The various functional blocks will be described in the following sections of this document.

## **2. Global Clock**

The TB will receive an external global clock from the FCS. This clock will be generated on shore and distributed to all TBs on all Nestor floors via the FCS (see “Requirements and Justification for the Nestor Fiber Communications System (FCS)”, section 3.2). Specifically, the clock will be the word-rate clock used to keep both ends of each fiber synchronized. By using one common clock source on shore we ensure that all Nestor floors are operating on exactly the same clock frequency. This clock operates in the range of 50-70MHz. Each TB will also have a local oscillator that operates in the same frequency range. The default clock used by the TB when it is powered on will be the local oscillator. However, during normal, deployed, operations the TB can be instructed (via the FCS and the BCB) to switch to the external global clock. Whichever clock has been selected as the global clock will be used as the FPGA clock. It will also be sent to the NDMs over the TCB. All communications over that path will then be synchronous with this clock.

## **3. FIFO**

The incoming data from each of the 12 comparators will be stored in a FIFO. Data will shift through the FIFO at the same speed as the NDMs send data to this TB, i.e. approximately 50MHz. The FIFO must be deep enough to hold the data from the 235ns before the trigger is generated. At a speed of 50MHz, or 20ns per sample, this corresponds to a minimum of 12 samples. When a trigger is generated all 12 samples currently stored in the FIFO will be read out, and then 12 more to record the data from the 235ns after the trigger was generated. This will match the depth of the FIFO readout on the NDMs.

## **4. Trigger Generator**

The trigger generator is responsible for generating the global trigger. It will be implemented in the FPGA along with the other digital logic. The trigger is generated in two stages: first a multiplicity trigger is generated using the comparator signals and then this is combined with the LED trigger and any Synchronization trigger that has been received to make the global trigger. There will be a scaler for each multiplicity value to count how many times that multiplicity occurs.

### **4.1. Multiplicity Logic Unit (MLU)**

The multiplicity trigger is generated using the 12 stretched comparator signals from the 2 NDMs. A multiplicity trigger will be generated whenever the number of

stretched comparator signals that are on (i.e. “1”) is over a user-settable threshold. This is shown schematically in Figure 3.

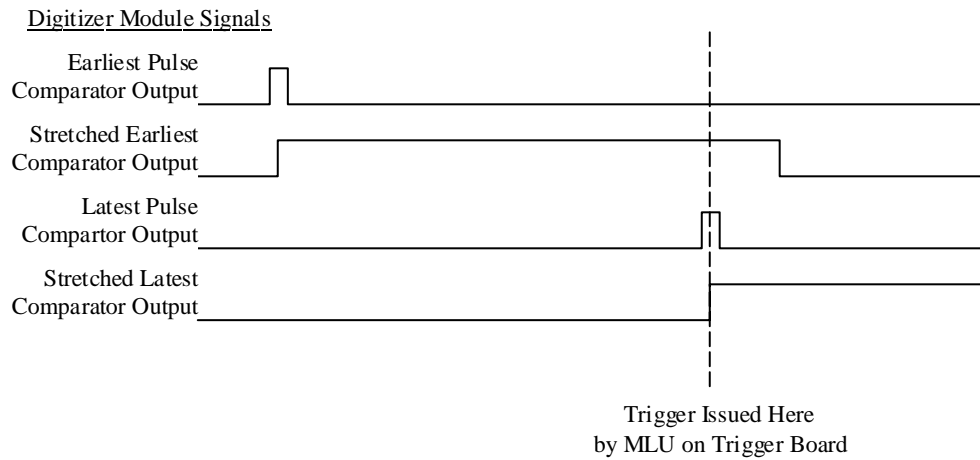


Figure 3: Generation of a 2-Fold Coincidence Multiplicity Trigger from an Early and Late Pulse

#### 4.2. Multiplicity Scalers

The multiplicity (i.e. the number of comparator signals that are on) can have one of 13 values between 0 and 12. 13 separate scalers will be implemented, one for each multiplicity value. Every tick of the clock the scaler corresponding to the current multiplicity value will be incremented by 1. All 13 scalers will be reset to zero after they have been read out, or on receipt of a command from the user. Each scaler will be at least a 28-bit counter. 28-bits are sufficient for one counter to increment at 50 MHz for 2.5 seconds (e.g., if the multiplicity is always 0 then scaler 0 will increment every time).

#### 4.3. Global Trigger

There will be 3 signals that contribute to the global trigger: the multiplicity trigger, the LED trigger from the SCS and the Synchronization trigger from the FFD. The three will be combined, i.e. OR'ed together, to create a global trigger. The trigger word will then be distributed to the time stamp counter and the operation control logic within this TB. It will also be sent to the two NDMs over the TCB to initiate readout there too. Once a global trigger has been issued the trigger generator will enter a “busy” state in which generation of further global triggers is blocked until the current event has been fully digitized and readout.

#### 4.4. Busy Logic

The generation of new triggers will be disabled when the system is busy. This can happen in one of 3 circumstances: the TB could be busy digitizing and reading out its

data from a previous trigger, the TB could have received a “busy” signal from either of the two NDMs indicating that it is unable to process new triggers or the memory could be full so there is nowhere to store any new data. Under normal conditions only the first two circumstances should actually occur. Typically the NDMs will stay busy for longer than the TB after receiving a trigger since they have more data to read out and store.

## 5. Time Stamp Counter

For monitoring purposes the TB will have a Time Stamp Counter that operates in exactly the same way as the Time Stamp Counter on the NDMs (see “Implementation of the Nestor Digitizer Module (NDM)”, section 4).

## 6. Memory

The TB will contain enough local memory to store up to 1.5 hours worth of triggered and scaler data. Table 2 lists the components of each event type and their size, assuming the scalers are read at a 1 Hz rate. The total comes to around 10 Mbytes which can easily fit into a commercial 64 Mbytes SIMM (DRAM).

	Triggered Events	Scaler Data	Total
Components	<b>Header:</b> 10 bytes <b>Time Stamp:</b> 7 bytes (51 bit counter) <b>FIFO:</b> 36 bytes (12 comparator signals, 24 time samples, 1 bit/sample) <b>Trigger Word:</b> 1 byte	<b>Header:</b> 10 bytes <b>Time Stamp:</b> 7 bytes (51-bit counter) <b>Scaler:</b> 52 bytes (13 scalers, 4 bytes/scaler)	
Event Size	54 bytes	69 bytes	
Event Rate	30 Hz	1 Hz	
Memory used in 1.5 hours	8.75 Mbytes	0.4 Mbytes	~ 10 Mbytes

Table 2: Event Sizes in the TB

The TB memory will be controlled by the operation control logic in the FPGA. The rate at which new data is written to the DRAM is approximately 1.7 Kbytes/second; 54 bytes/event at a 30 Hz event rate. This is much slower than the DRAM access time, which is approximately 80 Mbytes/second. It will therefore be possible to interleave the read and write operations so they do not interfere with each other, in the same way as for the NDM memory. As a result writing new events will not need to be suspended while currently stored events are read out so this memory will have no effect on the dead-time of the TB.

## **7. Dead-time**

The expected dead-time of the TB is negligible compared to the 10ms requirement (see “Requirements and Justification for the Nestor Trigger Board (TB)”, section 4.9). Of the processes involved in storing data from a triggered event (reading the FIFOs, saving the current time stamp and writing all the data into the memory) the memory access is expected to be the limiting time factor. Commercial DRAM can typically be accessed at 80 Mbytes/second (8 bytes at 10 MHz). At this rate it would take just 0.7  $\mu$ s to write one event of 54 bytes into the memory. The inherent dead-time of a TB operating without NDMs is therefore expected to be approximately 0.7  $\mu$ s, leading to a maximum steady-state event rate of 1.4 MHz.

## **8. Operation Control**

The operation control logic is responsible for configuring the TB, reading out the data, storing it and communicating with the FCS over the BCB. In response to commands from the FCS it will select the global clock source (local oscillator or external clock), set the multiplicity thresholds and set any other user-settable feature of this module. If this TB is configured to be the FFD master it will generate a synchronization trigger and distribute that trigger to its own Trigger Generator as well as to all the other floors via the FFD. When a global trigger is generated this logic component will gather the data from the 12 FIFOs, the time stamp and all the trigger information, form it into one data block and store it in the local, on-board, memory. At a user-settable rate the operation control logic will also read and clear all the scalers and store that data in the on-board memory. Finally, and again under control of the FCS, it will extract blocks of data from the local memory and pass them to the FCS over the BCB.

## **Glossary**

BCB	Backplane Communications Bus
FCS	Fiber Control System
FFD	Floor-to-Floor Daisy-Chain
MLU	Multiplicity Logic Unit
NDM	Nestor Digitizer Module
Nestor	Neutrino Extended Submarine Telescope with Oceanographic Research
PMT	Photomultiplier Tube
SCS	Slow Controls System
TB	Trigger Board
TCB	Trigger Communications Bus
TTS	Transit Time Spread